REMARKS

Upon entry of this amendment, claims 1-5, 8-13, 16-28, 30, 32, 33, 35-40 and 42-54 are all the claims pending in the application. Claims 6, 7, 14, 15, 29, 31, 34 and 41 have been canceled by this amendment, and claims 44-54 have been added as new claims. No new matter has been added.

I. Objections to the Specification

The Examiner has objected to the specification for the reasons set forth on page 3 of the Office Action. In particular, the Examiner indicated that the abstract should be a single paragraph within a range of 50 to 150 words. By this amendment, Applicants have modified the abstract so as to address the Examiner's objection. Applicants note that the specification also includes editorial amendments that have been made for grammatical and general readability purposes. No new matter has been added.

In view of the foregoing, Applicants respectfully request that the Examiner reconsider and withdraw the objection to the specification.

II. Objection to the Claims

The Examiner has objected to claims 3 and 4 for the reasons set forth on page 2 of the Office Action. By this amendment, Applicants note that claims 3 and 4 have each been amended by replacing the term "are" with the term "is", thereby addressing the Examiner's objection. Accordingly, Applicants respectfully request that the objections to the claims be reconsidered and withdrawn.

III. Claim Rejections under 35 U.S.C. § 112, second paragraph

The Examiner has rejected claims 2, 30 and 34 under 35 U.S.C. §112, second paragraph as being indefinite. By this amendment, Applicants have modified claims 2 and 30 based on the Examiner's comments so as to overcome the above-noted rejection. Regarding claim 34, as noted above, this claim has been canceled by this amendment.

In view of the foregoing, Applicants kindly request that the rejections under 35 U.S.C. 112, second paragraph be reconsidered and withdrawn.

IV. Claim Rejections under 35 U.S.C. § 101

The Examiner has rejected claims 1-43 under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Regarding claims 1-4, 8-12, 16-22, 24-27, 30, 32, 33, 42 and 43, the Examiner indicated in the Office Action that these claims are directed to only software modules, and are therefore non-statutory. By this amendment, Applicants note all of these claims are now drawn to a "computing device" including a "processor" and a "compiler apparatus". Thus, as each of the above-noted claims includes at least a "processor", Applicants respectfully that such claims clearly cannot be considered to be directly solely to software modules. Accordingly, Applicants respectfully request that the rejection under 35 U.S.C. 101 be reconsidered and withdrawn.

Regarding claims 5, 13, 23, 28 and 35, the Examiner indicated in the Office Action that these claims include only non-functional descriptive material. Applicants respectfully disagree.

For example, claim 5 recites, in part, a computer-readable medium on which a source program is recorded, wherein the source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language (1) not to allocate a specific array data to a global memory region and (2) to allocate the specific array data to the global memory region.

MPEP §2106.01 discusses the guidelines for determining whether or not a computer-related invention is patentable subject matter under 35 U.S.C. §101. This section of the MPEP indicates that "[d]escriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material".

In particular, as explained in MPEP 2106.01, "non-functional descriptive material" includes music, literary works and a compilation or mere arrangement of data, whereas "functional descriptive material" includes data structures which impart functionality when employed as a computer component (where the phrase "data structure" is defined as a physical or

logical relationship among data elements, designed to support specific data manipulation functions).

Applicants respectfully submit that claim 5 clearly includes data structures which impart functionality when employed as a computer component. For example, as noted above, claim 5 indicates that the source program includes at least one of descriptions for directing a compiler to perform a particular function regarding array data and the global memory region. As such, because a specific data manipulation function is being supported (i.e., whether or not to allocate a specific array to the global memory region), Applicants respectfully submit that claim 5 cannot merely be considered the same as "music" or a "literary work", which clearly do not impart any type of functionality.

In view of the foregoing, Applicants respectfully submit that claim 5 is directed to "functional descriptive material", and therefore, kindly request that the rejection under 35 U.S.C. 101 be reconsidered and withdrawn.

Regarding claim 13, Applicants note that claim 13 recites, in part, a computer readable recording medium on which a source program is recorded, wherein the source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language program (1) not to perform the optimization by software pipelining of a specific loop processing, (2) to perform optimization that removes a prolog portion and an epilog portion by software pipelining of the specific loop processing, and (3) to perform optimization that does not remove the prolog portion and the epilog portion by software pipelining of the specific loop processing.

For at least similar reasons as discussed above with respect to claim 5, Applicants respectfully submit that claim 13 is directed to statutory subject matter under 35 U.S.C. 101. Accordingly, Applicants kindly request that the rejection be reconsidered and withdrawn.

Regarding claim 23, Applicants note that this claim recites, in part, a computer-readable recording medium on which a source program is recorded, wherein the source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language program (1) to perform the optimization by loop unrolling of a specific loop

processing, (2) not to perform the optimization by loop unrolling of a specific loop processing, and (3) to require a guarantee on the number of iterations of a specific loop processing.

For at least similar reasons as discussed above with respect to claim 5, Applicants respectfully submit that claim 23 is directed to statutory subject matter under 35 U.S.C. 101. Accordingly, Applicants kindly request that the rejection be reconsidered and withdrawn.

Regarding claim 28, Applicants note that this claim recites, in part, a computer-readable recording medium on which a source program is recorded, wherein the source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language program (1) to make an "if" conversion to a specific "if" structure sentence and (2) not to make the "if" conversion to the specific "if" structure sentence.

For at least similar reasons as discussed above with respect to claim 5, Applicants respectfully submit that claim 28 is directed to statutory subject matter under 35 U.S.C. 101. Accordingly, Applicants kindly request that the rejection be reconsidered and withdrawn.

Regarding claim 35, Applicants note that this claim recites, in part, a computer-readable recording medium on which a source program is recorded, wherein the source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language program (1) to require a guarantee on alignment of data that a pointer variable of argument shown by the name of a specific variable indicates and (2) to require a guarantee on alignment of data that a local pointer variable shown by the name of a specific variable indicates.

For at least similar reasons as discussed above with respect to claim 5, Applicants respectfully submit that claim 35 is directed to statutory subject matter under 35 U.S.C. 101. Accordingly, Applicants kindly request that the rejection be reconsidered and withdrawn.

Regarding claims 36-40, the Examiner indicated that these claims were directed to computer listings per se. By this amendment, Applicants have amended claims 36-40 such that each of these claims is directed to a "computer readable medium having a program stored thereon". As such, Applicants respectfully submit that claims 36-40 are not directed to computer listings per se, and therefore, kindly request that the Examiner reconsider and withdraw the rejection under 35 U.S.C. 101.

V. Claim Rejections under 35 U.S.C. § 102

Claims 14-17, 29-34 and 41-43 were rejected under 35 U.S.C. § 102(b) as being anticipated by Stallman (Richard M. Stall, Using and Proting the GNU Compiler Collection for GCC 3.1).

Regarding claims 14, 15, 29, 31 and 41 as noted above, these claims have been canceled by this amendment.

Regarding claim 16, Applicants note that this claim recites that the directive acquisition unit detects a directive for performing the optimization by loop unrolling of a specific loop processing in the source program, and wherein the optimization unit performs the optimization by loop unrolling of loop processing that is an object of the directive detected by the directive acquisition unit. Applicants respectfully submit that Stallman does not disclose or suggest such features.

With respect to Stallman, Applicants note that this reference discloses the use of an option "-funroll-loops" for performing an optimization of loop unrolling only for loops whose number of iterations can be determined at compile time or run time (see page 55, lines 38-42). In this regard, Applicants point out that the "-funroll-loops" of Stallman is an option, not a pragma. A pragma, in contrast to an option, is treated as an individual directive.

Accordingly, in Stallman, while the object to be optimized is the loops whose number of iterations can be determined, Applicants respectfully submit that Stallman does not disclose or suggest that an object to be optimized is the loop designated by the directive.

In this regard, as noted above, claim 16 recites that the directive acquisition unit detects a directive for performing the optimization by loop unrolling of a specific loop processing in the source program, and wherein the optimization unit performs the optimization by loop unrolling of loop processing that is an object of the directive detected by the directive acquisition unit.

As noted above, because Stallman merely discloses that the object to be optimized is the loops whose number of iterations can be determined, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious at least the above-noted features

recited in claim 16. Accordingly, Applicants submit that claim 16 is patentable over Stallman, an indication of which is kindly requested.

Regarding claim 17, Applicants note that this claim recites that the directive acquisition unit detects a directive for not performing the optimization by loop unrolling of a specific loop processing in the source program, wherein the optimization unit restrains the optimization by loop unrolling of loop processing that is an object of the directive detected by the directive acquisition unit.

For at least similar reasons as discussed above with respect to claim 16, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious such features.

In particular, Applicants note that because Stallman merely discloses that the object to be optimized is the loops whose number of iterations can be determined, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious the features of directive acquisition unit that detects a directive for not performing the optimization by loop unrolling of a specific loop processing in the source program, wherein the optimization unit restrains the optimization by loop unrolling of loop processing that is an object of the directive detected by the directive acquisition unit.

In view of the foregoing, Applicants respectfully submit that claim 17 is patentable over Stallman, an indication of which is kindly requested.

Regarding claim 30, Applicants note that this claim recites that the directive acquisition unit acquires a directive for alignment of array data of a specific type together with a directive for translating the source program, wherein the optimization unit allocates all the array data of the specific type declared in the source program in the memory region so that its head address matches the alignment. Applicants respectfully submit that Stallman does not disclose or suggest such features.

In particular, Applicants note that while Stallman discloses an optimization for an individual array data (see section 5.33 of Stallman beginning on page 177), that Stallman does not disclose or in any way suggest that an optimization unit allocates all the array data of the

specific type declared in the source program in the memory region so that its head address matches the alignment, as recited in claim 30.

In view of the foregoing, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious all of the features recited in claim 30. Accordingly, Applicants submit that claim 30 is patentable over Stallman, an indication of which is kindly requested.

Regarding claim 32, Applicants note that this claim recites that the directive acquisition unit detects a designation of alignment of data that a pointer variable of argument shown by the name of a specific variable indicates in the source program, wherein the optimization unit performs the optimization assuming that the data that is an object of designation detected by the directive acquisition unit is allocated in the memory region by the designated alignment. Applicants respectfully submit that Stallman does not disclose or suggest such features.

In the Office Action, the Examiner has indicated that Stallman discloses the above-noted features at page 182, lines 16-20. Applicants respectfully disagree.

In particular, Applicants note that the section of Stallman cited by the Examiner merely indicates that "[i]f you declare or use arrays of variables of an efficiently-aligned type, then it is likely that your program will also be doing pointer arithmetic."

Thus, while Stallman discloses the use of "pointer arithmetic" in conjunction with arrays of variables of an efficiently-aligned type, Applicants respectfully submit that such disclosure in no way whatsoever suggests the detection of a designation of alignment of data that a pointer variable of argument shown by the name of a specific variable indicates in the source program, as recited in claim 32.

In this regard, as mentioned above, Applicants note that a pragma is a directive that a user can arbitrarily designate in the source program and can specify not all loops or variables, but an individual loop or variable, whereby optimization can be performed only for the data that is an object of the directive designated by the user. With respect to claim 32, Applicants note that Stallman does not disclose or even remotely suggest the use of a pragma related to the alignment of data indicated by a pointer variable.

In view of the foregoing, Applicants submit that claim 32 is patentable over the cited prior art, an indication of which is kindly requested. Claim 42 depends from claim 32 and is therefore considered patentable at least by virtue of its dependency.

Regarding claim 33, Applicants note that this claim recites that the directive acquisition unit detects a designation of alignment of data that a local pointer variable shown by the name of a specific variable indicates in the source program, and wherein the optimization unit performs the optimization assuming that the data that is an object of designation detected by the directive acquisition unit is allocated in the memory region by the designated alignment.

For at least similar reasons as discussed above with respect to claim 32, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious such features. Accordingly, Applicants submit that claim 33 is patentable over the cited prior art, an indication of which is kindly requested. Claim 43 depends from claim 33 and is therefore considered patentable at least by virtue of its dependency.

VI. Claim Rejections under 35 U.S.C. § 103(a)

A. Claims 35 and 40 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1).

Claim 35 recites that a source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language program (1) to require a guarantee on alignment of data that a pointer variable of argument shown by the name of a specific variable indicates and (2) to require a guarantee on alignment of data that a local pointer variable shown by the name of a specific variable indicates.

For at least similar reasons as discussed above with respect to claims 32 and 33, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious the above-noted combination of features. Accordingly, Applicants submit that claim 35 is patentable over Stallman, an indication of which is kindly requested.

Claim 40 recites the feature of an optimization unit operable to perform optimization by generating a sequence of machine language instructions following the acquired directive, wherein

the optimization unit performs optimization by allocating data in a memory region following a directive when the optimization unit acquires the directive on alignment of the array data to be allocated in a memory region.

For at least similar reasons as discussed above with respect to claims 32 and 33, Applicants respectfully submit that Stallman does not disclose, suggest or otherwise render obvious the above-noted combination of features recited in claim 40. Accordingly, Applicants submit that claim 40 is patentable over Stallman, an indication of which is kindly requested.

B. Claims 18-23 and 38 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1) in view of Geva (U.S. 6,539,541).

Claim 18 recites that the directive acquisition unit detects a designation of the number of iterations of specific loop processing in the source program, wherein the optimization unit performs optimization of loop processing that is an object of the designation detected by the directive acquisition unit based on the designated number of iterations. Applicants respectfully submit that the combination of Stallman and Geva does not teach or suggest the above-noted features.

As discussed above with respect to claims 16 and 17, Stallman discloses the use of an option "-funroll-loops" for performing an optimization of loop unrolling only for loops whose number of iterations can be determined at compile time or run time (see page 55, lines 38-42). In this regard, Applicants point out that the "-funroll-loops" of Stallman is an option, not a pragma. As discussed previously, a pragma, in contrast to an option, is treated as an individual directive.

Accordingly, in Stallman, while the object to be optimized is the loops whose number of iterations can be determined, Applicants respectfully submit that Stallman does not disclose or in any way suggest that optimization of loop processing can be an object of the designation detected by a directive acquisition unit, as recited in claim 18. Further, Applicants respectfully submit that Geva fails to cure this deficiency of Stallman.

In view of the foregoing, Applicants respectfully submit that the above-noted features recited in claim 18 are not disclosed, suggested or rendered obvious by the cited prior art. Accordingly, Applicants submit that claim 18 is patentable over the cited prior art, an indication of which is kindly requested. Claims 19-22 depend from claim 18 and are therefore considered patentable at least by virtue of their dependency.

Regarding claim 23, Applicants note that this claim recites that the source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language program (1) to perform the optimization by loop unrolling of a specific loop processing, (2) not to perform the optimization by loop unrolling of a specific loop processing, and (3) to require a guarantee on the number of iterations of a specific loop processing.

For at least similar reasons as discussed above with respect to claim 18, Applicants respectfully submit that the combination of Stallman and Geva does not teach, suggest or otherwise render obvious the above-noted features recited in claim 23. Accordingly, Applicants submit that claim 23 is patentable over the cited prior art, an indication of which is kindly requested.

Regarding claim 38, Applicants note that this claim recites that an optimization unit is operable to perform optimization by generating a sequence of machine language instructions following an acquired directive, wherein the optimization unit performs optimization by loop unrolling following a directive when the directive acquisition unit acquires the directive on the optimization by loop unrolling.

For at least similar reasons as discussed above with respect to claim 18, Applicants respectfully submit that the combination of Stallman and Geva does not teach, suggest or otherwise render obvious the above-noted features recited in claim 38. Accordingly, Applicants submit that claim 38 is patentable over the cited prior art, an indication of which is kindly requested.

C. Claims 1-5 and 36 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1) in view of Nakamura (Nakamura et al., Architecture and Compiler Co-Optimization for High Performance Computing).

Claim 1, as amended, recites that a global memory region is specified by a head address and a displacement, wherein the head address is indicated by a value stored in a register, wherein the displacement is within a range of the global memory region that can be accessed by one instruction, and wherein the range is determined based on a type and a size of an object. Applicants respectfully submit that the combination of Stallman and Nakamura does not teach or suggest such features.

Regarding Stallman, as noted above, this reference discloses the use of an option "-funroll-loops" for performing an optimization of loop unrolling only for loops whose number of iterations can be determined at compile time or run time. Applicants respectfully submit, however, that Stallman does not disclose or in any way suggest the above-noted features recited in amended claim 1 drawn to the displacement being within a range of the global memory region that can be accessed by one instruction.

Regarding Nakamura, Applicants note that this reference discloses a software controllable memory (SCM) which can be configured dynamically by software and which is integrated into an architecture called SCIMA (see section 1 of Nakamura) to fill the performance gap between a processor and a memory speed. In Nakamura, a directive based compiler is disclosed, where "hints" for optimization are given by users as directives (see section 3 of Nakamura).

Thus, while Nakamura discloses the use of a directive based compiler, Applicants respectfully submit that Nakamura does not disclose or even suggest the above-noted features recited in amended claim 1 of a global memory region that is specified by a head address and a displacement, wherein the head address is indicated by a value stored in a register, wherein the displacement is within a range of the global memory region that can be accessed by one instruction, and wherein the range is determined based on a type and a size of an object.

In view of the foregoing, Applicants respectfully submit that the cited prior art does not teach, suggest or otherwise render obvious all of the features recited in amended claim 1. Accordingly, Applicants submit that claim 1 is patentable over the cited prior art, an indication of which is kindly requested. Claims 2-4 depend from claim 1 and are therefore considered patentable at least by virtue of their dependency.

Regarding claims 5 and 36, Applicants note that each of these claims has been amended so as to recite that the global memory region is specified by a head address and a displacement, wherein the head address is indicated by a valued stored in a register, wherein the displacement is within a range of the global memory region that can be accessed by one instruction, and wherein the range is determined based on a type and a size of an object.

For at least similar reasons as discussed above with respect to claim 1, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious such features. Accordingly, Applicants submit that claims 5 and 36 are patentable over the cited prior art, an indication of which is kindly requested.

D. Claims 6-8 and 13 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1) in view of PPCREF (PPCREF Project - High level design, Software Pipelining and Branch Optimization).

Regarding claims 6 and 7, Applicants note that both of these claims have been canceled.

Regarding claim 8, Applicants note that this claim recites that the directive acquisition unit detects a directive for not performing the optimization by software pipelining of a specific loop processing in the source program, wherein the optimization unit restrains the optimization by software pipelining of loop processing that is an object of the directive detected by the directive acquisition unit.

With respect to the above-noted features recited in claim 8, as described previously, a pragma is a directive that a user can arbitrarily designate in the source program and can specify not all loops or variables, but an individual loop or variable in the source program.

Regarding the combination of Stallman and PPCREF, Applicants note that Stallman merely discloses the use of a directive format in gcc '#pragma implementation' which controls inline functions in an explanation for options, and PPRCEF only discloses how to conduct software pipelining.

Thus, Applicants respectfully submit that neither reference teaches the ability to detect a directive for not performing the optimization by software pipelining of a specific loop processing in the source program, or an optimization unit that restrains the optimization by software pipelining of loop processing that is an object of the directive detected by the directive acquisition unit, as recited in claim 8.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious all of the features recited in claim 8. Accordingly, Applicants submit that claim 8 is patentable over the cited prior art, an indication of which is kindly requested.

Regarding claim 13, Applicants note that this claim recites that a source program includes at least one of descriptions for directing a compiler that translates the source program into a machine language program (1) not to perform the optimization by software pipelining of a specific loop processing, (2) to perform optimization that removes a prolog portion and an epilog portion by software pipelining of the specific loop processing, and (3) to perform optimization that does not remove the prolog portion and the epilog portion by software pipelining of the specific loop processing.

For at least similar reasons as discussed above with respect to claim 8, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious such a combination of features. Accordingly, Applicants submit that claim 13 is patentable over the cited prior art, an indication of which is kindly requested.

E. Claims 9-12 and 37 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC

3.1) in view of PPCREF (PPCREF Project - High level design, Software Pipelining and Branch Optimization) in further view of Granston (U.S. 6,892,380).

Regarding claim 9, Applicants note that this claim recites that the directive acquisition unit detects a directive for performing the optimization by software pipelining that removes a prolog portion and an epilog portion of a specific loop processing in the source program, wherein the optimization unit performs the optimization by software pipelining of loop processing that is an object of the directive detected by the directive acquisition unit whenever possible to remove the prolog portion and the epilog portion.

In the Office Action, the Examiner has recognized that Stallman and PPCREF do not teach or suggest the above-noted features. The Examiner, however, has applied Granston and has taken the position that this reference cures the deficiencies of Stallman and PPCREF. Applicants respectfully disagree.

In particular, regarding Granston, Applicants note that this reference is directed to a method of software pipelining which provides the ability to omit the epilog (see col. 4, lines 19-26). Thus, while Granston discloses that the epilog can be removed, Applicants note that such disclosure, even when considered in conjunction with the disclosure of Stallman and PPCREF, does not teach or suggest that a directive is detected for performing optimization by software pipeline that removes a prolog portion and an epilog portion of a specific loop processing in the source program. In this regard, it is noted that the Examiner has not addressed in the Office Action the feature of the "prolog portion" being removed.

Further, with respect to the above-noted feature of a directive for performing the optimization of a specific loop processing, the Examiner has indicated in the Office Action that it is well know in the computer art that the compilation directive can be added to selectively optimize a specific loop (see Office Action at page 26). In view of the Examiner's comments, it is clear that the Examiner is taking Official Notice with respect to such a feature.

Accordingly, if the Examiner maintains the rejection, Applicants request that the Examiner cite to a reference in support of this position as required by MPEP §2144.03.

In this regard, even if the Examiner cites to a reference in support of such a position, Applicants submit that it would not have been obvious to one of ordinary skill in the art to modify Stallman to provide such a feature because Stallman is explicitly directed to an optimization which is performed on all loops whose number of iterations can be determined.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not disclose, suggest or otherwise render obvious the above-noted combination of features recited in claim 9. Accordingly, Applicants submit that claim 9 is patentable over the cited prior art, an indication of which is kindly requested

Regarding claim 10, Applicants note that this claim recites that the directive acquisition unit detects a directive for performing the optimization by software pipelining that does not remove the prolog portion and the epilog portion of a specific loop processing in the source program, wherein the optimization unit performs the optimization by software pipelining of loop processing that is an object of the directive detected by the directive acquisition unit whenever possible not to remove the prolog portion and the epilog portion.

In the Office Action, the Examiner has rejected claim 10 based on the same rationale as claim 9. As is clear from the language in claim 10, however, claim 10 recites that a directive is detected for performing the optimization by software pipelining that does not remove the prolog portion and the epilog portion. In contrast to this claim language, the Examiner has relied on Granston for the teaching of the removal of an epilog portion.

As such, it is clear that the Examiner's rejection of claim 10 is improper because the claim language has not been properly addressed. Therefore, Applicants submit that claim 10 is patentable over the cited prior art, an indication of which is kindly requested.

Further, for at least similar reasons as discussed above with respect to claim 9, Applicants respectfully submit that there would be no reason to modify the cited prior art references by providing a compilation directive that can selectively optimize a specific loop. As discussed above, if the Examiner maintains that such a feature is well known in the art, Applicants request that the Examiner cite a reference in support of such a position in accordance with MPEP 2144.03, and if such a reference is cited, Applicants request that the Examiner explain with

particularity why one of ordinary skill in the art would modify the Stallman reference in such a manner.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious all of the features recited in claim 10. Accordingly, Applicants submit that claim 10 is patentable over the cited prior art, an indication of which is kindly requested.

Regarding claim 11, Applicants note that this claim recites that the directive acquisition unit detects a designation of the number of iterations of specific loop processing in the source program, wherein the optimization unit performs optimization of loop processing that is an object of the designation detected by the directive acquisition unit based on the designated number of iterations.

In the Office Action, the Examiner has recognized that Stallman and PPCREF do not teach or suggest the above-noted features. The Examiner, however, has applied Granston and has taken the position that this reference cures the deficiencies of Stallman and PPCREF. Applicants respectfully disagree.

In particular, regarding Granston, Applicants note that this reference is directed to a method of dynamic software pipelining which describes that a typical FOR loop, when the loop begins, "n" represents the number of desired iterations. In Granston, it is noted that the dynamic software pipelining is performed by a processor, and therefore, while "n" may represent a desired number of iterations, this actual value of iterations is not fixed. In contrast, Applicants note that claim 11 is directed to static software pipelining performed by a compiler.

As such, Applicants respectfully submit that while Granston discloses that "n" represents a desired number of iterations, that such disclosure does not in any way suggest the ability to detect a designation of the number of iterations of specific loop processing in the source program, wherein an optimization unit performs optimization of loop processing that is an objection of the designation detected by the directive acquisition unit based on the designated number of iterations.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not disclose, suggest or otherwise render obvious all of the features recited in claim 11. Accordingly, Applicants submit that claim 11 is patentable over the cited prior art, an indication of which is kindly requested. Claim 12 depends from claim 11 and is therefore considered patentable at least by virtue of its dependency.

F. Claims 24-28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1) in view of Prata (Stephen Prata, C Primer Plus, Third Edition).

Regarding claim 24, Applicants note that this claim has been amended to recite that the optimization unit performs optimization on an "if" conversion following a directive when the directive acquisition unit acquires the directive on the "if" conversion, wherein the "if" conversion is rewriting an if construction in the source program to a conditional instruction without using a branch instruction.

In the Office Action, the Examiner has recognized that Stallman does not teach or suggest the use of an "if" conversion, and has applied Prata for the teaching of #ifdef and #endif directives, both of which are common directives in C language.

Thus, while Prata discloses the use of common directives such as #ifdef and #endif, Applicants respectfully submit that Prata does not include any disclosure whatsoever regarding rewriting an if construction in the source program to a conditional instruction without using any branch instruction, as recited in amended claim 24.

In view of the foregoing, Applicants respectfully submit that the cited prior art references do not disclose, suggest or otherwise render obvious all of the features recited in amended claim 24. Accordingly, Applicants submit that claim 24 is patentable over the cited prior art, an indication of which is kindly requested. Claims 25-27 depend from claim 24 and are therefore considered patentable at least by virtue of their dependency.

Regarding claim 28, Applicants note that this claim recites that the source program includes at least one of descriptions for directing a compiler that translates the source program

into a machine language program (1) to make an "if" conversion to a specific "if" structure sentence and (2) not to make the "if" conversion to the specific "if" structure sentence, wherein the "if" conversion is rewriting an if construction in the source program to a conditional instruction without using a branch instruction.

For at least similar reasons as discussed above with respect to claim 24, Applicants respectfully submit that the cited prior art references do not teach, suggest or otherwise render obvious the above-noted features recited in claim 28. Accordingly, Applicants respectfully submit that claim 28 is patentable over the cited prior art, an indication of which is kindly requested.

VII. New Claims

Claims 44-54 have been added as new claims.

Regarding claims 44 and 45, Applicants note that claim 44 depends from claim 16, and claim 45 depends from claim 17. Accordingly, Applicants submit that claims 44 and 45 are patentable at least by virtue of their dependency.

Regarding claim 46, Applicants note that this claim recites that the directive acquisition unit detects a designation of the number of iterations of specific loop processing in the source program, wherein the optimization unit performs optimization of loop processing that is an object of the designation detected by the directive acquisition unit based on the designated number of iterations.

For at least similar reasons as discussed above with respect to claim 11, Applicants respectfully submit that the cited prior art references do not disclose, suggest or otherwise render obvious the above-noted features. Accordingly, Applicants submit that claim 46 is patentable over the cited prior art, an indication of which is kindly requested. Claims 47-51 depend from claim 46 and are therefore considered patentable at least by virtue of their dependency.

Regarding claims 52-54, Applicants note that these claims depend from claims 8-10, respectively. Accordingly, Applicants submit that claims 52-54 are patentable at least by virtue of their dependency.

VIII. Conclusion

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may best be resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

Respectfully submitted,

Hajime OGAWA et al.

By: Kenneth W. Fields

Kenneth W. Fields
Registration No. 52,430
Attorney for Applicants

KWF/jjv
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
July 26, 2007